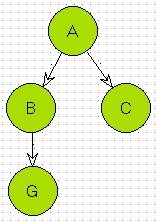
**Parte 1:**

**1-**

**a)**

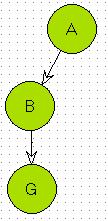
**i)**

|  |  |
| --- | --- |
| 1. | A |
| 2. | B |
| 3. | C |
| 4. | G |

****

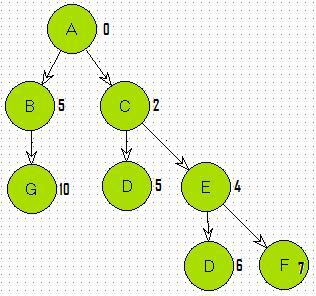
**ii)**

|  |  |
| --- | --- |
| 1. | A |
| 2. | B |
| 3. | G |

****

**iii)**

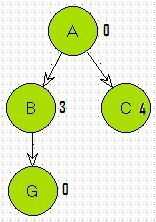
|  |  |
| --- | --- |
| 1. | A |
| 2. | C |
| 3. | D |
| 4. | E |
| 5. | D |
| 6. | F |
| 7. | G |

****

**iv)**

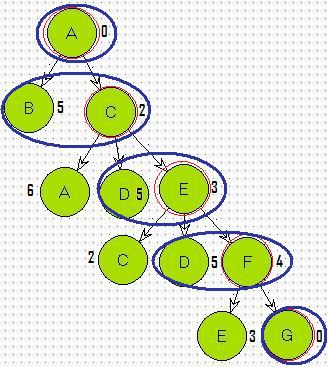
**h2:**

|  |  |
| --- | --- |
| 1. | A |
| 2. | B |
| 3. | C |
| 4. | G |



**h3:**

|  |  |
| --- | --- |
| 1. | A |
| 2. | B |
| 3. | C |
| 4. | A |
| 5. | D |
| 6. | E |
| 7. | C |
| 8. | D |
| 9. | F |
| 10. | E |
| 11. | G |

****

**NOTA:**

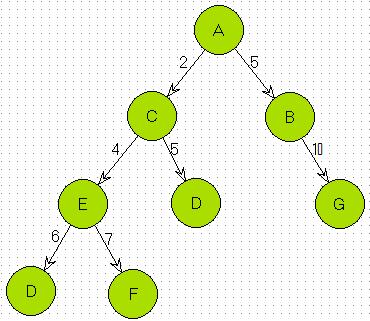
Azul -> Nós a serem considerados

Vermelho -> Nós expandidos

Neste caso apenas se considerou os nós que nunca foram visitados, chegando assim à solução, para o algoritmo ser realmente ganancioso todos os nós seriam considerados e assim entraria-se num loop infinito. Baseado nestes slides:<http://cee.uma.pt/edu/iia/acetatos/iia-Procura%20Informada.pdf>

**b)**

h1: Solução: ABG



h2:

Solução: ACDG

h3:

Solução: ACDG

h4:

Solução: ACDG

**c)**

PLargura: Não é óptimo.

PProfundidade: Não é óptimo.

Custo Uniforme: Óptimo.

A\*: Óptimo.

**2-**

**3-**

**Parte 2:**

**a)**

**r** - factor de ramicação médio;

**p** - profundidade máxima;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Complexidade Temporal | Complexidade Espacial | Completo | Óptimo |
| Primeiro em Profundidade | rp | r\*p | Não | Não |
| Primeiro em Largura | rp+1 | rp+1 | Sim | Sim |
| Aprofundamento Progressivo | rp | r\*p | Sim | Sim |

Exemplo:

r=10

p(solução)=12

Nó=byte

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Complexidade Temporal | Complexidade Espacial | Vantagens | Situação a Utilizar |
| Primeiro em Profundidade | 1012  1000000000000 | 10\*12=120 | - Requer pouca memória  - O nó objectivo pode vir a ser encontrado sem  examinar a árvore por completo. | - No caso de haver pouca margem para armazenamento de memória |
| Primeiro em Largura | rp+1 | rp+1 | -Encontra sempre a solução se existir.  -Encontra a melhor em comprimento do passo. | - Para grandes profundidades e àrvores desconhecidas. |
| Aprofundamento Progressivo | 1012  1000000000000 | 10\*12=120 |  | - É uma excelente opção para problemas em  que somos obrigados a recorrer a um método cego. |

**b)** (retirado da resolução do exercício 2.1 das aulas práticas (exercício dos baldes))

% :-use\_module(library(lists)). %Sicstus

%% primeiro em profundidade

resolve\_pp:-

inicio(Ei),

primeiro\_prof(Ei, L, [Ei]),

reverse(L,L2),

nl, escreve(L2).

primeiro\_prof(Ea, [Ea],\_):- objectivo(Ea).

primeiro\_prof(Ea, [Ea|R], Eants):-

sucessor(Ea, Eseg,Accao),

not(member(Eseg, Eants)),

primeiro\_prof(Eseg, R, [Eseg|Eants]).

escreve(L):- L=[Ei|\_],

write('estado inicial : '), write(Ei),nl,

escreve1(L).

escreve1([\_]):- !.

escreve1([Ea,Eseg|R]):-

sucessor(Ea,Eseg,Accao),write(Accao),write(' : '),

write(Eseg), nl,

escreve1([Eseg|R]).

Predicado sucessor é a lista de nós da árvore a percorrer.

// Meu

primeiro\_profundidade(Objectivo;Lista\_Dados;Profundidade;Solução):-

pp\_aux(Objectivo;Lista\_Dados;Profundidade;0;Solução;[]).

pp\_aux(Obj;[Obj|Tail];Prof;Prof\_actual;Solução;Sol):-

Prof\_actual<=Prof,

append(Sol,Obj,Solução).

pp\_aux(Obj;[Head|Tail];Prof;Solução;Sol):-

// Meu

**c)** Considerando “c” o custo real, as heurísticas só são admissíveis se:

**i)** h3 = h1 + h2;

É admissível se h1 < c e h2 < c e h1+h2 < c. **SIM**

**ii)** h4 = min(h1; h2);

É admissível se h1 < c e h2 < c, logo min(h1;h2) < c. **SIM**

**iii)** h5 = max(h1; h2);

É admissível se h1 < c e h2 < c, logo max(h1;h2) < c. **SIM**

**iv)** h6 = h1 \* h2;

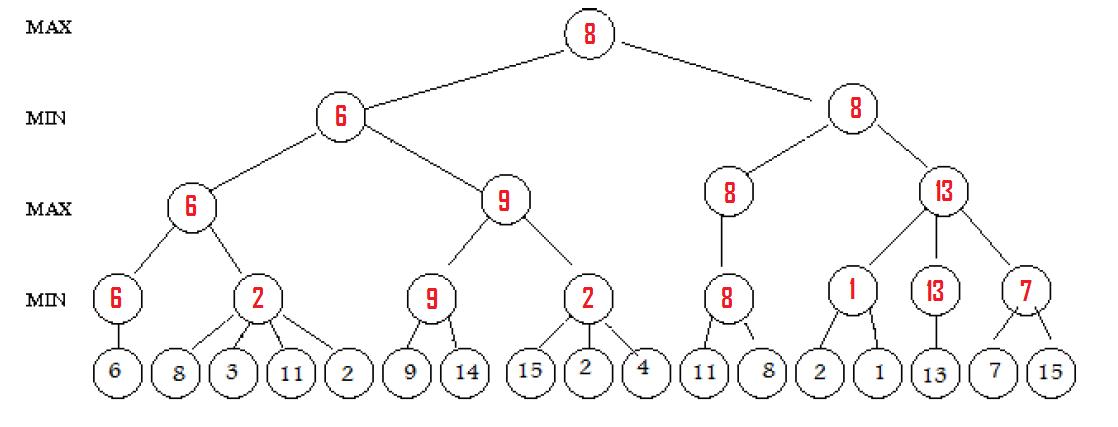
É admissível se h1 < c e h2 < c e h1\*h2 < c. **SIM**

**v)** h7 = w\*h1 + (1-w)\*h2 com w Є [0, 1].

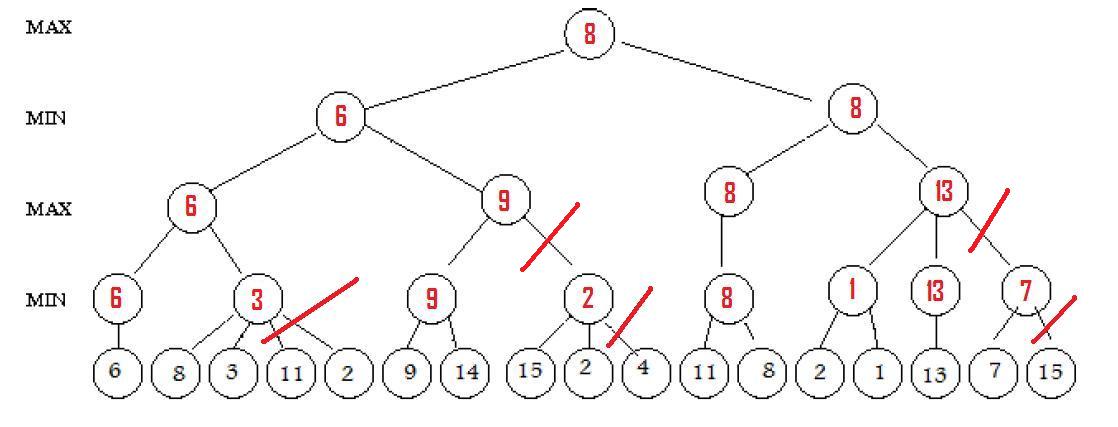
É admissível se:

* h1 < c;
* h2 < c;
* w < 1;
* w > 0;
* w\*h1 + (1-w)\*h2 < c; **SIM**

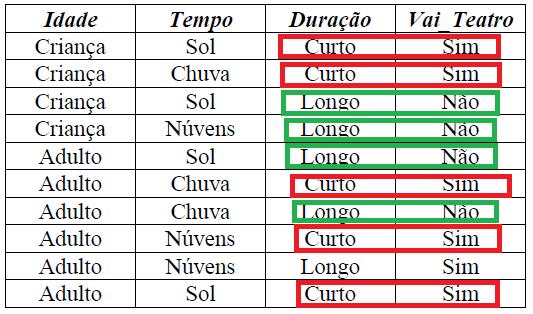
**d)** Sem cortes:



Com cortes:



**e)**

****

Sem efectuar cálculos, é possível afirmar que o “nó raíz” da árvore é o atributo **duração**, pois a sua entropia é a menor. Quando a duração é *curta*, *Vai\_Teatro=SIM* sempre, quando a duração é *longa* não se pode afirmar nada, mas apenas um em cinco afirma *Vai\_Teatro=NÃO*.

**f)**

**g)**

**h)** Regras:

* R1: Se tempo está mau ou disposição é má então Sr X não vai ao jogo (FC=0.9)
* R2: Se Sr X acha que vai chover então tempo está mau (FC=0.7)
* R3: Se Sr X acha que vai chover e meteorologia diz que vai chover então disposição é má (FC=0.8)
* R4: Se tempo está mau então disposição é má (FC=0.9)

Sabendo que a meteorologia diz que vai chover (FC=0.8) e o Sr.X acha que vai chover (FC=0.9), qual a conclusão que o Sistema Pericial retira sobre a ida do Sr.X ao jogo?

**1. TM** ou **DM ->** FC(**NV**)=0.9

**2. AC** -> FC(**TM**)=0.7

**3. AC** e **MC** -> FC(**DM**)=0.8

**4. TM** -> FC(**DM**)=0.9

FC(**MC**)=0.8

FC(**AC**)=0.9

3;2;4;1

**3.** FC[**D**isposição **M**á] = 0.8\*min(**AC**;**MC**)= 0.8\*min(0.9;0.8)=0.8\*0.8=0.64

**2.** FC[**T**empo **M**au] = 0.7\*0.9= 0.63

**1.** FC[**N**ao **V**ai ao Jogo] = 0.9\*max(**TM**;**DM**) = 0.9\*max(0.63;0.64) = 0.9\*0.64= 0.576

**FC[Vai ao Jogo]= 1-0.576 = 0.424**